Clustering the Unknown - The Youtube Case

Amit Dvir[1], Angelos K. Marnerides[2], Ran Dubin[1], Nehor Golan[1]

[1]*Cyber Innovation Center, Department of Computer Science, Ariel University, Israel*

[2]*InfoLab21, School of Computing and Communications, Lancaster University, UK*

amitdv,dubin,nehorgo@ariel.ac.il, angelos.marnerides@lancaster.ac.uk

***Abstract*—Recent stringent end-user security and privacy requirements caused the dramatic rise of encrypted video streams in which YouTube encrypted traffic is one of the most prevalent. Regardless of their encrypted nature, meta-data derived from such traffic flows can be utilized to identify the title of a video, thus enabling the classification of video streams into a single video title using a given video title set. Nonetheless, scenarios where no video title set is present and a supervised approach is not feasible, are both frequent and challenging. In this paper we go beyond previous studies and demonstrate the feasibility of clustering unknown video streams into subgroups although no information is available about the title name. We address this problem by exploring Natural Language Processing (NLP) formulations and Word2vec techniques to compose a novel statistical feature in order to further cluster unknown video streams. Through our experimental results over real datasets we demonstrate that our methodology is capable to cluster 72 video titles out of 100 video titles from a dataset of 10,000 video streams. Thus, we argue that the proposed methodology could sufficiently contribute to the newly rising and demanding domain of encrypted Internet traffic classification.**

***Index Terms*—Encrypted Traffic, Video Title, Clustering, YouTube**

1. Introduction

Several recent pieces of work have demonstrated that meta-data derived from encrypted traffic may be adequately used in the context of classifying video streams [1], [2]. Naturally, such a capability is considered quite useful in terms of group user habits to enable adaptive QoE mechanisms [3]–[6] but is also crucial when identifying trends of interest to intelligence agencies serving the aspect of cyber threat surveillance [7]–[12].

A common scenario of classifying meta-data to infer the video a user is watching is when encrypted video streams are clustered based on training labels built by a pool of video titles. For instance, Dubin et al. in [7], [8], compared and introduced a machine learning (ML)-based methodology for classifying encrypted HTTP adaptive video titles. The applicability of the proposed scheme was demonstrated in the scenario where an external attacker could identify a video title from the video streams as long as the title of the video stream is from a given video title set. Nonetheless, in case the title of the video stream is not from the given set the proposed algorithm classifies the stream as unknown video. The classification process by Dubin et al. [8], first examined YouTube traffic and showed that the HTTP DASH protocol, as being the main YouTube mechanism for video streams, uses the standard HTTP byte range mode. Thus, each segment can request a different byte range depending on the client-side playout buffer levels as well as network conditions. However, due to anticipated varying network conditions, even if the user watches the same video title with the same quality varying bandwidths are likely to persist as presented in Fig. 1 [8].

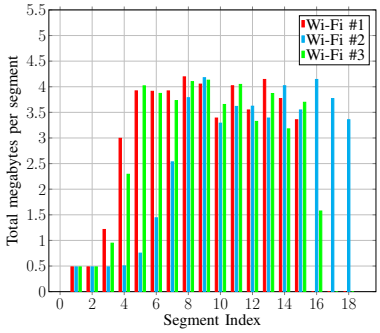

Figure 1. Total megabytes per segment of three Youtube downloads, same video title, same quality and different Wi-Fi. [8]

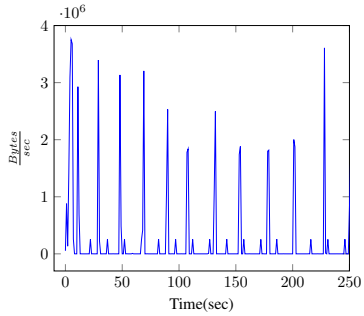

Figure 2. Typical example of YouTube encrypted traffic via browser (Chrome), BPP is the sum of bytes between two Off periods where Off mean when $\frac{bytes}{sec} = 0$ [8]

Subsequently, with the use of the On/Off behavior of YouTube [13] Dubin et al. [8] defined the Bit Per Peak

(BPP) feature that we also use in this work, which in practice is the sum of bytes between two Off periods as shown in Fig. 2. As evidenced in Fig 2, the downloads from the Youtube server have Off periods ($\frac{bytes}{sec} = 0$) and between two Off periods there are peaks. From each peak we create BPP, sum of the bytes. Hence, it was feasible to describe a given video stream as a set of BPPs.

To respond to the need of clustering the unknown, we first implemented the BPP values in Dubin et al. [7], [8] as the feature vector (which is the vector of the sum of the bytes) of a video title. Subsequently we exploited principles of Natural Language Processing (NLP) and developed a language from network traffic features, BPPs, per video stream. Note that, BPPs (network traffic on/off patterns) have distinct properties that can be used as features for distinguishing between different titles. Finally, we utilized the K-Means algorithm [14] to cluster the encrypted network traffic video streams.

Therefore the main contributions of this paper are:

- The first study to assess encrypted traffic clustering under an NLP formulation.
- A methodology where a novel NLP-based feature composition process is included in order to aid the task of clustering unknown video streams when video title information is not available.

The remainder of this paper is organized as follows. Section 2 provides a summary of related work and highlights the novelty behind our scheme, whereas section 3 describes the dataset used in this work. Section 4 is dedicated on presenting the methodology employed in this work. Section 5 discusses the evaluation undertaken in this paper, and, finally, section 6 concludes and summarizes this paper.

2. Related Work

In general, several studies aimed to cluster groups of encrypted traffic using only network statistics [15]–[17]. Erman et al [15] utilised transport layer statistics in order to cluster encrypted network traffic. Their results indicate that clustering is indeed a useful technique for traffic identification and achieved to classify application protocols (e.g. http, p2p, smtp). Bacquet et al. [16] used Multi-Objective Genetic Algorithm (MOGA) for feature selection and cluster count optimization to cluster application protocols from encrypted traffic. In their work, they used the flow parameters in addition to transport layer parameters. Hochst et al. [17] assessed mobile applications and presented a novel approach to unsupervised traffic flow classification based on flows statistic whereas clustering was based on a neural auto encoder. However, in contrast to our work, most of past studies aim to cluster the encrypted traffic into corresponding application layer protocols with minimal success on explicitly looking at encrypted video traffic clustering based on the video title.

Under a common mindset with our herein proposed work, Li [12] presented Silhouette; a real-time, lightweight video classification method that only uses statistics for video title identification; Stikkelorum [11] used state machines to match video segments and reward segments in order to identify the video title in the encrypted Youtube video stream. Moreover, Reed and Kranch [9] used direct network observations to identify Netflix videos streaming, and Schuster et al. [10] noted that video streams are uniquely characterized by their burst patterns, such that by implementing a Convolutional Neural Network (CNN) they could accurately identify them.

Nonetheless, regardless of the insightful outputs of the aforementioned studies they all were restricted to classifying video streams from well-known video title sets. Hence, to date there is no method that explicitly attempted to classify or at least cluster video streams that were identified as unknown.

3. Dataset Description

As already mentioned, this work assesses the clustering of encrypted video streams with no a-priori knowledge or access on a known set of labeled video titles. Therefore, we collected a dataset of 10,000 YouTube video streams. The dataset was downloaded using a real-world Internet connection over a period of several months under different real-world network conditions using Chrome as a browser. In this study we decided to use the Chrome browser for two reasons, since it is the most popular browser with growing popularity and due to the fact that the YouTube On/Off behavior exists in all the browsers [13]. We downloaded the entire data-set from several network connections and conditions using the Selenium web automation tool [18] with ChromeDriver [19] for the crawler. This simulates a user video download to mimic normal user behavior. We did not assume any prior knowledge about how many different flows existed per stream. Our resulted dataset contains 100 video titles, where each video title was downloaded 100 times. We used popular YouTube videos from different categories such as news and sports. In each download stream, we utilise the auto mode of the YouTube player where the player decides which quality representation to download based on estimations of the client's network conditions.

Lets define j as a video title and vs_j as video stream of video title j. When a user watches a video title j, the stream data which is the packets downloaded from the server to the client define as the video stream. Due to the fact that we download every video title several times (e.g. 100 times), we define $vs_{m,j}$ as a video stream number m of title j where in our case $m \in [1 - 100]$ (streams) and $j \in [1-100]$ (titles). A summary of the dataset parameters can be seen in Table 1

TABLE 1. DATASET PARAMETERS

Number of video titles	100
Number of video streams per video title	100
Total number of video streams	10,000 (100*100)
Video types	news, sports, nature, video trailers, GoPro
Browser	Chrome
Automation	Selenium
YouTube player	Auto Mode

4. Clustering the Unknown

In this section we present our methodology as demonstrated in Fig. 3. The first module is the preprocessor module where we eliminate retransmissions and audio packets from the video stream as described in Section 4.1. Within the second module, the encrypted traffic data are processed in order to create the BPPs for each video stream by combining the packets (full packet, header and payload) between two Off periods within the observed peaks.

In the third module the Word2vec algorithm is employed in order to generate a language from network traffic features and compose the novel features that are based on the previously created BPPs. As evidenced in Fig. 3, the fourth module is in charge of triggering the K-means algorithm, in order to cluster the collected video streams under an unsupervised fashion by using the NLP-based features created in the third module as described in Section 4.2. Finally, the last module is dedicated at evaluating the clustering procedure undertaken in the fourth module. We evaluate our algorithm by calculating the number of different video titles have in each bin, where optimal algorithm will have video streams from only one video title in each bin.

4.1. Preprocessing

First, based on the well-known five tuple representation: protocol (TCP/UDP), src IP, dst IP, src port, dst port we divide the encrypted traffic into flows. Then, to decide whether the flow is a YouTube stream we use either the Service Name Indication (SNI) field in the Client Hello message (e.g. googlevideos.com) or machine learning techniques [20], [21]. Then we can remove audio packets; i.e., bursts below 400kB, since video traffic bursts are much larger. Note that the audio data and the video data can be found in the same flow and in some cases we cannot distinguish between them. Finally, to avoid the influence of network conditions as much as possible we eliminate TCP re-transmissions using a TCP stack [22].

4.2. Feature Composition

The direct use of raw data in a machine learning method is problematic since raw data tend to be non-structured and normally contain redundant information. A natural and prominent solution is to initially perform feature extraction and then feature selection to construct a structured representation that in parallel is tailored to the specific problem domain.

Unsupervised classification learning methods learn a classification function from a set of unlabeled examples by using heuristics. Based on the herein reported problem domain, we cluster unknown video streams to determine whether there are streams of the same video title, without a-priori knowledge of the video title. This requires using unsupervised learning for classification. One example of an unsupervised algorithm is the K-Means algorithm [14], which approximates a division of the dataset into n distinct clusters of equal variance where each cluster is described by the mean. Hence, K-Means attempts to find the mean values that minimize the intra-cluster sum of squares; i.e., the squared Euclidean distance between all samples within a cluster and its respective mean.

The core principle of NLP is to understand the meaning of a word. Although the general, human-like ability to understand language remains elusive, certain methods have been successful in capturing similarities between words. Recently, neural-network based approaches in which words are embedded into a low dimensional space have been proposed by various authors [23], [24]. These models represent each word as a d-dimensional vector of real numbers. Vectors that are close to each other were shown to be semantically related by Mikolov et al. [25], [26]. Mikolov et al., proposed a skip-gram with a negative-sampling training method which enables an efficient embedding of words that achieves striking results on various linguistic tasks.

In the skip-gram model, unlike most previously used neural network architectures for learning word vectors, the training process does not involve dense matrix multiplications. Rather, the model aims to learn high quality vector representations of words from large amounts of unstructured text data. Generally speaking, for a sentence of n words $w_1, ..., w_n$, contexts of a word w_i come from a window of size win around the word: $C_i(win) = w_{(i-win)}, ..., w_{(i-1)}, w_i, w_{(i+1)}, ..., w_{(i+win)}$, where win is a parameter and C_i is the context. The window size win can be either static or dynamic; if dynamic the parameter win denotes the maximal window size and for each word in the corpus, a window size win is sampled uniformly from $1, ..., n$.

Goldberg [27] found that larger windows induce embeddings that are more topical or associative, thus improving their performance on analogy test sets, whereas smaller windows induce more functional and synonymic models leading to better performance on similarity test sets. The window size effect on the encrypted network traffic clustering is unknown and is discussed here for the first time.

In our problem domain, clustering video titles, we define a BPP as a word w_i (convert the integer value to a word). Moreover, a vector of BPPs converted from integers (number of bits between On/Off period) into set of words and become a sentence of n words. Our feature transformation works as follows. For each video stream we extract the BPPs and transform them into a vector of strings instead of a vector of integers. Then, we use Word2vec with Skip-gram and empirical re-iterations selecting the best window sizes based on the clustering algorithm ability to cluster the same titles in a single cluster bin as our metric. The Word2vec window size defines the maximum distance between the current and predicted word within a sentence.

5. Performance Evaluation

We aimed to cluster the video streams into bins where each bin is a different video title, when no information is available about this title. Therefore, if the algorithm clusters video streams with the same title into the same bin we increase the value of the bin by one. In the end, the value of the bin emphasis the number of different title clustered by the algorithm. An optimal algorithm will cluster video streams of the same title

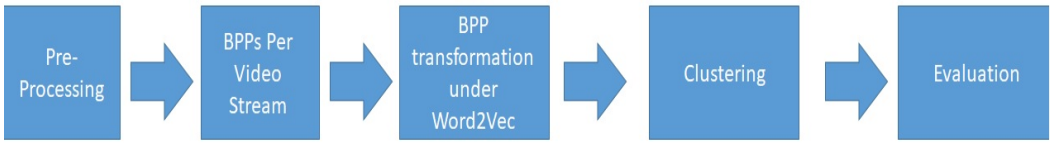

Figure 3. The 5 modules of the proposed methodology for clustering encrypted video streams.

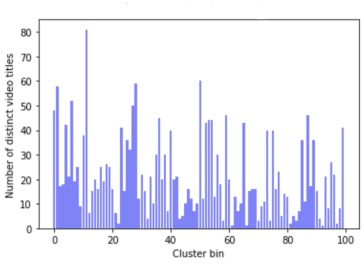

Figure 4. Number of distinct video titles, BPP values as features, K-means, $k = 100$

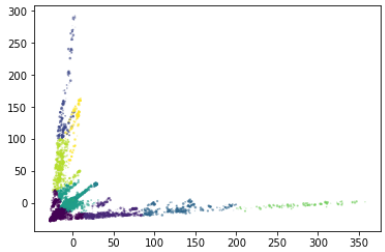

Figure 5. BPP values as features, K-means + PCA, $k = 10$

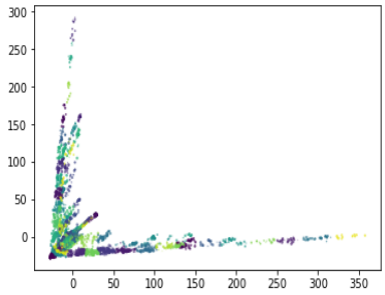

Figure 6. BPP values as features, K-means + PCA, $k = 100$

into one bin and therefore the value of each bin will be one. Otherwise, if the clustering puts different video titles in the same bin, for each title we increase the value of the bin by one. For example, if in a bin we clustered $vs_{1,1}, vs_{5,1}, vs_{9,1}, vs_{80,1}, vs_{8,7}, vs_{15,7}, vs_{97,7}$ (4 video streams of video title 1 and 3 video streams of video title 7) so the value of this bin will be, 2 which is the number of different video title. To summarized, in order to assess the clustering algorithm we calculate the number of different video titles have in each bin (we sum up the value of j in each bin).

First we assessed the performance of our scheme using the BPPs as vector of integers. The findings can be found in Fig. 4. The figure indicates that bin '0' has a value of 48, which means 48 distinct video titles clustered to this center whereas bin '98' has a value of 2 which means only streams from 2 different videos were clustered into this bin. Unfortunately, only a single bin (bin '92') separated a single title whereas the others have many titles.

An analysis of the feature vectors revealed high variance of the BPP values and the number of BPPs in the video stream. This may have led to the well known dimension problem where the number of dimensions tends to infinity, and the distance between any two points in the dataset converges. Thus, the maximum distance and minimum distance between any two points of the dataset will be the same [28]. To reduce the high dimensionality in our datasets we employed the Principal Components Analysis (PCA) method. Figures 5 -7 depict the PCA with K-means. They show that a separation is possible but it remains unclear how many titles are in each cluster bin.

This result underscored the need for an alternative data transformation where the relationship between the assessed features is derived. We used Word2vec [25], [26] to transform the numerical BPP feature into a word. Hence, each video stream became to a vector of words.

Subsequent to language creation from the network traffic features, we explored the influence of the Word2vec window size. Figure 8 illustrates Word2vec with k-means' ability to separate the data with different window sizes. As evidenced from the figure, the best window size is 82 where this value separated the titles into 72 distinct bins out of 100 (72 bins with value equal to one).

Figure 9 presents the results of clustering with K-means using the Word2vec. The plot shows that most of the titles have a value of 1, thus indicating that only streams of this title cluster to the bin and the others in most cases have a low value. Note that, in some cases where the value of the bin is small (e.g. 2-4) it means that the algorithm cluster video streams of several video title into the same bin. Therefore, in those cases our algorithm achieve also good results.

Figures 10 - 11 present a deep analysis of the resulted bins (i.e. clusters). In practice, the resulted plots show the name (unique and short name we gave to each movie due to lack of space in the figure) of the video titles clustered to a given bin and the number of streams of the same

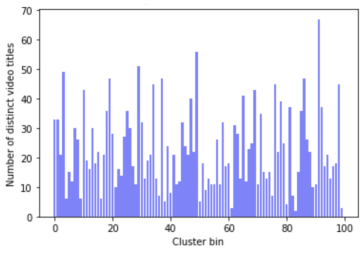

Figure 7. Number of distinct video titles, BPP values as features, K-means + PCA, $k = 100$

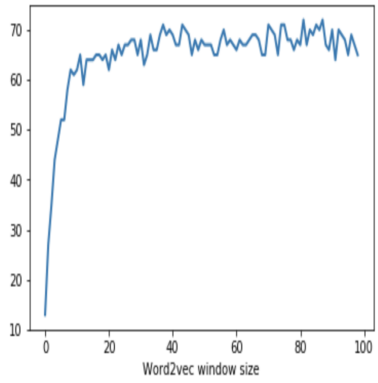

Figure 8. Optimizing the Word2vec window size, number of distinct clusters Vs Word2vec window size

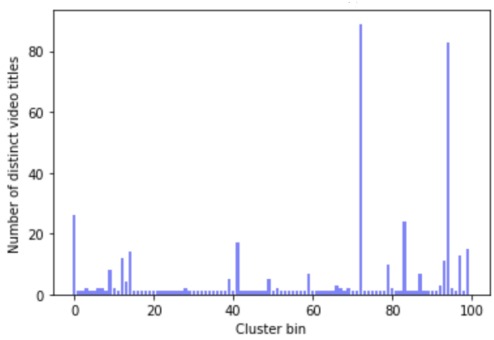

Figure 9. Number of distinct video titles, BPP values as words, Word2vec + K-means, $win = 82$, $k = 100$

title clustered. The large font indicates the number of video streams of the same title clustered to this bin. For instance, in the case of bin '9' in Fig. 10, we can see that several titles clustered to this bin where in the case of bin '21' in Fig. 11, only two titles were clustered. In Fig. 11, only video streams of title "Party" and "String" clustered, where many of the video streams clustered to the video title "Party" but only a few to the title "Single".

In general, Fig.11 depicts the greater picture in terms of the novelty in this paper. As explained earlier, our methodology manages to demonstrate the feasibility of clustering unknown video streams into subgroups when no information is available about the title name. We can see that in most of the cases, 72, we clustered the video streams of a video title into one bin. Even in the cases when we fail to cluster streams of only one video title to the same bin, our algorithm clustered streams of few titles (e.g. 2-5).

The clustering purity [29] measures the average value

Figure 10. Bin number 9 - an example of bad clustering, with many titles in the same bin

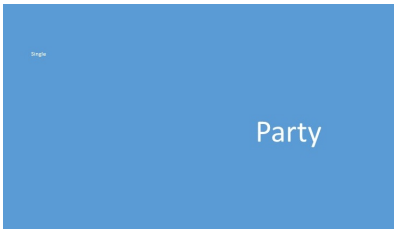

Figure 11. Bin number 21 - an example of good clustering, only two titles are in the same bin. The titles are "Party", large font (represent many streams of the title) and "single", very small font (represent few streams of the title)

of the ratios between the largest class in each cluster to the size of the cluster. Our algorithm, K-means with Word2vec and window size 82, achieved a clustering purity of 0.566.

The average Silhouette value [30] is a measurement that lies in the range $[-1, 1]$ and indicates the tightness of the clustering separation. For each data point, its Silhouette value is measured (a value in the range $[-1, 1]$) and this value which indicate how close is it to it's neighboring clusters. When -1 indicates that the point is close to a neighboring cluster, and value of 1 is preferred and indicating the no relations to the neighbor clusters. Our algorithm, K-means with Word2vec with window size 82, have average Silhouette value of 0.453.

6. Conclusions

This work addressed the challenging aspect of encrypted video stream classification with no a-priori knowledge of video title names and assessed explicitly the YouTube case. We have proposed a novel synergistic method where features derived by packet-based volume features such as the sum of bits on YouTube's On/Off feature (i.e. BPPs) are treated under an NLP paradigm to compose a meaningful feature set. By utilising the proposed feature set we demonstrate over real datasets that we may efficiently and accurately cluster unknown encrypted video streams under an unsupervised clustering scheme. We argue that to the best of our knowledge this is the first work illustrating clustering of encrypted video stream titles using an NLP approach. We finally claim, that the herein reported findings may establish strong foundations towards the composition of scalable and accurate classification schemes for encrypted traffic.

References

[1] Flow clustering using machine learning techniques. In Chadi Barakat and Ian Pratt, editors, *Passive and Active Network Measurement*, pages 205–214. Springer Berlin Heidelberg, 2004.

[2] R. Lychev, S. Jero, A. Boldyreva, and C. Nita-Rotaru. How secure and quick is quic? provable security and performance analyses. In *Symposium on Security and Privacy, (SP)*, pages 214–231, May 2015.

[3] Bogdan Iacob. Streaming video detection and qoe estimation in encrypted traffic. *Network*, 1, 2018.

[4] R. Dubin, A. Dvir, O. Pele, O. Hadar, I. Raichman, and O. Trabelsi. Real time video quality representation classification of encrypted http adaptive video streaming-the case of safari. In *IEEE Digital Media Industry and Academic Forum*. IEEE, 2016.

[5] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau. Internet video delivery in youtube: from traffic measurements to quality of experience. In *Data Traffic Monitoring and Analysis*. 2013.

[6] O. Oyman and S. Singh. Quality of experience for http adaptive streaming services. *IEEE Communications Magazine*, 50(4):20–27, April 2012.

[7] R. Dubin, A. Dvir, O. Pele, and O. Hadar. I know what you saw last minute-encrypted http adaptive video streaming title classification. In *Black Hat*, 2016.

[8] R. Dubin, A. Dvir, O. Pele, and O. Hadar. I know what you saw last - encrypted http adaptive video streaming title classification. *IEEE Transactions on Information Forensics and Security*, 12(12):3039–3049, Dec 2017.

[9] A. Reed and M. Kranch. Identifying https-protected netflix videos in real-time. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, CODASPY '17, pages 361–368, 2017.

[10] R. Schuster, V. Shmatikov, and E. Tromer. Beauty and the burst: Remote identification of encrypted video streams. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1357–1374. USENIX Association, 2017.

[11] Melcher Stikkelorum. I know what you watched: Fingerprint attack on youtube video streams. 2017.

[12] Feng Li, Jae Won Chung, and Mark Claypool. Silhoue e–identifying youtube video flows from encrypted traffic. In *ACM SIGMMWorkshop on Network and Operating Systems Support for Digital Audio and Video*, 2018.

[13] P. Ameigeiras, J. Ramos-Muoz, J. Navarro-Ortiz, and J. M. Lpez-Soler. Analysis and modelling of youtube traffic. *Transactions on Emerging Telecommunications Technologies*, 23(4):360–377, 2012.

[14] C. Ding and X. He. K-means clustering via principal component analysis. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML, pages 29–36, 2004.

[15] Jeffrey Erman, Martin Arlitt, and Anirban Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, MineNet '06, pages 281–286, 2006.

[16] Carlos Bacquet, A. Nur Zincir-Heywood, and Malcolm I. Heywood. Genetic optimization and hierarchical clustering applied to encrypted traffic identification. In *2011 IEEE Symposium on Computational Intelligence in Cyber Security, CICS 2011, Paris, France, April 12-13, 2011*, pages 194–201, 2011.

[17] J. Hochst, L. Baumgartner, M. Hollick, and B. Freisleben. Unsupervised traffic flow classification using a neural autoencoder. In *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, volume 00, pages 523–526, Oct. 2018.

[18] Selenium. Selenium automates browsers. http://www.seleniumhq.org/. Accessed: 2018-02-28.

[19] Chromedriver - webdriver for chrome. https://sites.google.com/a/chromium.org/chromedriver/. Accessed: 2018-03-28.

[20] P. Fu, L. Guo, G. Xiong, and J. Meng. Classification research on ssl encrypted application. In *Trustworthy Computing and Services*. 2013.

[21] G. L. Sun, Y. Xue, Y. Dong, D. Wang, and C. Li. An novel hybrid method for effectively classifying encrypted traffic. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[22] R. Dubin, O. Hadar, A. Noam, and R. Ohayon. Progressive download video rate traffic shaping using tcp window and deep packet inspection. In *WORLDCOMP*, 2012.

[23] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[24] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[27] Yoav Goldberg. A primer on neural network models for natural language processing. *J. Artif. Intell. Res.(JAIR)*, 57:345–420, 2016.

[28] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006.

[29] Junjie Wu, Hui Xiong, and Jian Chen. Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 877–886. ACM, 2009.

[30] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.